A Compositional Framework for Boolean Networks

H. Alkhudhayr[a,b], J. Steggles[a]

{h.alkhudhayr1, jason.steggles}@ncl.ac.uk

[a] School of Computing, Newcastle University, Newcastle upon Tyne, UK
[b] Faculty of Computing and Information Technology, King Abdulaziz University, Rabigh, Saudi Arabia

Abstract

Boolean networks are a widely used qualitative approach for modelling and analysing biological systems. However, their application is restricted by the well–known state space explosion problem which means that modelling large-scale, realistic biological systems is challenging. In this paper we set out to facilitate the construction and analysis of large scale biological models by developing a formal framework for the composition of Boolean networks. The compositional approach we present is based on merging entities between Boolean networks using a binary Boolean operator and we formalise the preservation of behaviour under composition using a notion of compatibility. We investigate characterising compatibility in terms of the composed models by developing a trace alignment property. In particular, we use a formalisation of the interference that can occur in a composed model to define an extended trace alignment property that we show completely characterises compatibility.

Keywords: Boolean network, model composition, behaviour preservation, interference

1. Introduction

Qualitative modelling techniques have become increasingly important in biology as the basis for developing formal techniques and tools for modelling, analysing and synthesizing biological systems [1, 2]. *Boolean networks* [3, 4] are a widely used qualitative approach based on representing the state of each biological entity (e. g. genes, proteins and other chemical signals) abstractly as a Boolean value, where 1 represents the entity is active and 0 that it is inactive. Entities then interact to regulate their states based on predefined next–state functions and the resulting dynamic behaviour gives rise to attractor cycles that can then be associated with biological phenomena. Entities can update their state either *synchronously*, where the state of all entities is updated simultaneously, or *asynchronously*, where entities update their state independently.

Boolean networks have been shown to allow a range of interesting biological analysis to be performed and have been widely considered in the literature (for example, see [5, 6, 7, 8, 2]). It seems clear that they have an important role to play in advancing our understanding and engineering capability of complex biological systems. However, one important issue that limits the scalable application of Boolean networks is the well–known state space explosion problem which means that modelling and analysing large-scale, realistic biological systems is challenging.

In this paper we set out to facilitate the construction and analysis of large scale biological models by developing a formal framework for the composition of (synchronous) Boolean networks. The compositional approach we present is based on composing Boolean networks by merging entities using a binary Boolean operator (there are 16 such binary Boolean operators to choose from [9]). We consider what it means to preserve the underlying behaviour of Boolean networks in a composed model and introduce the notion of *compatibility* to formalise this concept. It turns out that the compatibility property is problematic to check as it references the behaviour of the composed model and so we develop the *alignment* property which is able to predict whether the composition of Boolean networks is compatible based only on their individual behaviour. We show formally that the alignment property is sufficient to ensure compatibility when the underlying Boolean operator being used is idempotent and illustrate its application by presenting results about the compatibility of composing duplicate copies of a Boolean network.

While the alignment property is interesting it turns out that is not a necessary property for compatibility and so does not completely characterise behaviour preservation. We therefore investigate extending the alignment property by considering the behavioural interference that can occur when two Boolean networks are composed. We formalise this interference using a labelled state graph approach and then use this to define an extended alignment property called *weak alignment*. We show formally that weak alignment completely characterises compatibility when the underlying Boolean operator being used is idempotent and therefore provides an important scalable means of checking behaviour preservation in composed models.

The compositional framework developed here is supported by a prototype tool that automates the composition process and associated analysis.

We note that an initial version of this work which focused on the sufficient property of alignment appears in [10].

This paper is organized as follows. In Section 2 we provide a brief introduction to Boolean networks. In Section 3 we develop a compositional framework for Boolean networks and introduce the notion of *compatibility* to formalise the preservation of behaviour under composition. In Section 4 we investigate characterising compatibility and introduce the *alignment* property, a sufficient property for ensuring compatibility. In Section 5 we extend this alignment result to provide a complete characterisation of compatibility by modelling the interference that can occur within a composed model. Finally, in Section 6 we present some concluding remarks and discuss future work.

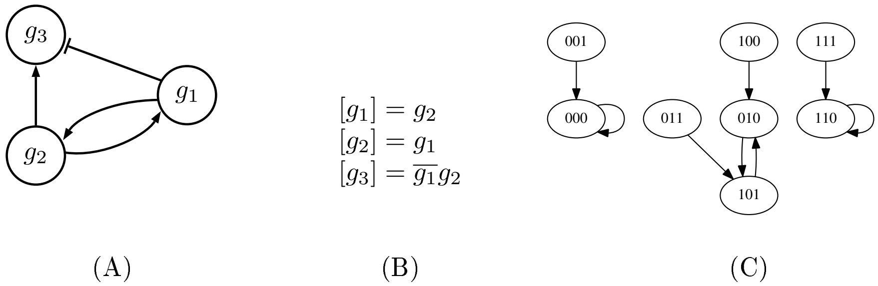

Figure 1: Example of a Boolean network $\mathcal{BN}_{Ex1}$ consisting of: (A) Wiring diagram; (B) Equational definition of next–state functions for $\mathcal{BN}_{Ex1}$; (C) Synchronous state graph.

2. Boolean Networks

Boolean networks [3, 4] are a widely used qualitative modelling approach for biological control systems (see for example [5, 6, 7, 8, 2]). In this section we introduce the basic definitions for Boolean networks needed in the sequel and provide illustrative examples.

A Boolean network consists of a set of regulatory entities $G = \{g_1, ..., g_n\}$ which can be in one of two possible states, either 1 representing the entity is active (e. g. a gene is expressed or a protein is present) or 0 representing the entity is inactive (e. g. a gene is not expressed or a protein is absent). The state of each entity is regulated by a subset of entities in the Boolean network and we refer to this subset as the *neighbourhood* of an entity (an entity may or may not be in its own neighbourhood). An entity updates its state by applying a logical *next–state function* to the current states of the entities in its neighbourhood.

We can define a Boolean network more formally as follows.

Definition 1. A *B*oolean Network $\mathcal{BN}$ is a tuple $\mathcal{BN} = (G, N, F)$ where:
i) $G = \{g_1, \ldots, g_n\}$ is a non-empty, finite set of entities;
ii) $N = (N(g_1), \ldots, N(g_n))$ is a tuple of neighbourhoods, such that $N(g_i) \subseteq G$ is the neighbourhood of g_i; and
iii) $F = (F(g_1), \ldots, F(g_n))$ is a tuple of next-state functions, such that the function $F(g_i) : \mathbb{B}^{|N(g_i)|} \to \mathbb{B}$ defines the next state of g_i.

As an example, consider the Boolean network $\mathcal{BN}_{Ex1} = (G_{Ex1}, N_{Ex1}, F_{Ex1})$ defined in Figure 1. It consists of three entities $G_{Ex1} = \{g_1, g_2, g_3\}$ with neighbourhoods $N_{Ex1}(g_1) = \{g_2\}$, $N_{Ex1}(g_2) = \{g_1\}$, and $N_{Ex1}(g_3) = \{g_1, g_2\}$. The next-state functions F_{Ex1} are defined equational in Figure 1.(B), where we use $[g_i]$ to represent the next state of an entity g_i.

A *global state* of a Boolean network $\mathcal{BN}$ with n entities is represented by a tuple of Boolean states $(s_1, ..., s_n)$, where $s_i \in \mathbb{B}$ represents the state of entity $g_i \in \mathcal{BN}$. Note as a notational convenience we often use $s_1...s_n$ to represent a global state $(s_1, ..., s_n)$. When the current state of a Boolean network is clear from the context we allow g_i to denote both the name of an entity and its corresponding current state. The state space of a Boolean network $\mathcal{BN}$, denoted $S_{\mathcal{BN}}$, is therefore the set of all possible global states $S_{\mathcal{BN}} = \mathbb{B}^{|G|}$.

The state of a Boolean network can be updated either *synchronously* [4, 11], where the state of all entities is updated simultaneously in a single update step, or *asynchronously* [12], where entities update their state independently. In the following we focus on the synchronous update semantics which has received considerable attention in the literature (see for example [3, 4, 5, 11, 13, 14]). Given two states $S_1, S_2 \in S_{\mathcal{BN}}$, let $S_1 \xrightarrow{\mathcal{BN}} S_2$ represent a *(synchronous) update step* such that S_2 is the state that results from simultaneously updating the state of each entity g_i using its associated update function $F(g_i)$ and the appropriate neighbourhood of states from S_1. As an example, consider the global state 011 for $\mathcal{BN}_{Ex1}$ (see Figure 1), where entity $g_1 = 0$, $g_2 = 1$, and $g_3 = 1$. Then $011 \xrightarrow{\mathcal{BN}_{Ex1}} 101$ is an update step in $\mathcal{BN}_{Ex1}$.

The sequence of global states through $S_{\mathcal{BN}}$ from some initial state is called a *trace*. Note that in the case of the synchronous update semantics such traces are deterministic and infinite. Given that the global state space is finite, this implies that a trace must eventually enter a cycle, known formally as an *attractor cycle* [4, 15]. Attractor cycles are important biologically where they are seen as representing different biological states or functions (e. g. different cellular types such as proliferation, apoptosis and differentiation [16]). We define a finite canonical representation for synchronous traces which specifies the infinite behaviour of a trace up to the first repeated state as follows. Let $S_0 \in S_{\mathcal{BN}}$ be a global state for $\mathcal{BN}$. A *trace* is a list of global states $\sigma(S_0) = \langle S_0, S_1, \ldots, S_{n+1}\rangle$ such that:
i) $S_i \xrightarrow{\mathcal{BN}} S_{i+1}$, for $0 \leq i \leq n$;
ii) $S_0, \ldots, S_n$ are unique states; and
iii) $S_{n+1} = S_i$ for some $i \in \{0, \ldots, n\}$.

The set of all traces $Tr(\mathcal{BN}) = \{\sigma(S) \mid S \in S_{\mathcal{BN}}\}$ therefore completely characterizes the behaviour of a Boolean network $\mathcal{BN}$ under the (synchronous) update semantics. For example, in $\mathcal{BN}_{Ex1}$ (see Figure 1) the trace $011 \xrightarrow{\mathcal{BN}_{Ex1}} 101 \xrightarrow{\mathcal{BN}_{Ex1}} 010 \xrightarrow{\mathcal{BN}_{Ex1}} 101 \xrightarrow{\mathcal{BN}_{Ex1}} \ldots$ is denoted by

$$\sigma(011) = \langle 011, 101, 010, 101\rangle$$

It can be seen that $\mathcal{BN}_{Ex1}$ has three attractors: two point attractors $\langle 000, 000\rangle$ and $\langle 110, 110\rangle$; and a cyclic attractor $\langle 101, 010, 101\rangle$.

The behaviour of a Boolean network can be concisely represented by a *state graph* in which the nodes are the global states and the edges are precisely the synchronous update steps allowed. We let $SG(\mathcal{BN}) = (\mathcal{S}_{\mathcal{BN}}, \xrightarrow{\mathcal{BN}})$ denote the state graph for a Boolean network $\mathcal{BN}$ under the synchronous trace semantics. As an example, consider the synchronous state graph $SG(\mathcal{BN}_{Ex1})$ for $\mathcal{BN}_{Ex1}$ presented in Figure 1.(C).

3. Compositional Framework

In this section we begin to develop our formal framework for composing Boolean networks based on the idea of merging entities using a binary Boolean connective (for example, conjunction). We consider what it means for the behaviour of an individual Boolean network in a composed model to be preserved and formalise this by introducing a notion of *compatibility*.

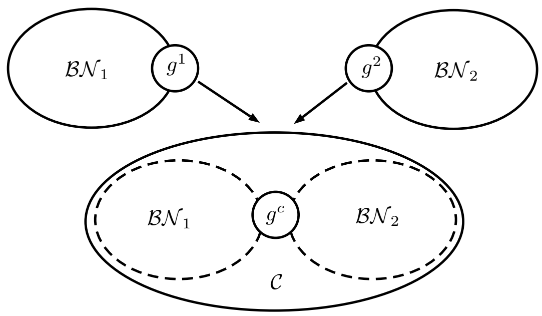

Figure 2: Pictorial representation of composing $\mathcal{BN}_1$ and $\mathcal{BN}_2$ to form a new Boolean network $\mathcal{C}$ by merging entities $g^1 \in \mathcal{BN}_1$ and $g^2 \in \mathcal{BN}_2$ into a new entity g^c.

In the sequel, let $\mathcal{BN}_1 = (G_1, N_1, F_1)$ and $\mathcal{BN}_2 = (G_2, N_2, F_2)$ be two Boolean networks such that $G_1 = \{g^1, g^1_1, ..., g^1_n\}$ and $G_2 = \{g^2, g^2_1, ..., g^2_m\}$ are disjoint sets of entities, for some $n, m \in \mathbb{N}$. We let $\odot$ represent an arbitrary binary Boolean operator chosen from the 16 possible binary Boolean operators available (see for example [9]).

We formally define the composition of two Boolean networks $\mathcal{BN}_1$ and $\mathcal{BN}_2$ using $\odot$ to merge the behaviour of entities (see Figure 2) as follows.

Definition 2. (Composition) Let $\mathcal{C}^{\odot}\left(\mathcal{BN}_1, \mathcal{BN}_2, g^1, g^2\right)$ denote the Boolean network constructed by composing $\mathcal{BN}_1$ and $\mathcal{BN}_2$ on entities g^1 and g^2 using $\odot$ defined as follows:

1. Entities: The finite set of entities $G = (G_1/\{g^1\}) \cup (G_2/\{g^2\}) \cup \{g^c\}$, where g^c denotes the new entity created by merging g^1 and g^2.

2. Neighbourhood: For any entity $h_i \in G$, the neighbourhood $N(h_i)$ is defined as follows:

$$N(h_i) = \begin{cases} N_1(h_i)[g^1/g^c], & \text{if } h_i \in G_1 \\ N_2(h_i)[g^2/g^c], & \text{if } h_i \in G_2 \\ N_1(g^1)[g^1/g^c] \cup N_2(g^2)[g^2/g^c], & \text{if } h_i = g^c \end{cases}$$

where $S[f/e]$ represents set S with all occurrences of element f replaced by e.

3. Functions: For any $h_i \in G$, the next-state function $F(h_i)$ is defined:

$$F(h_i) = \begin{cases} F_1(h_i), & \text{if } h_i \in G_1 \\ F_2(h_i), & \text{if } h_i \in G_2 \\ \mathcal{F}, & \text{if } h_i = g^c \end{cases}$$

Let $N_1(g^1) = \{l_1, ..., l_p\}$ if $g^1 \notin N_1(g^1)$, otherwise let $N_1(g^1) = \{g^1, l_1, ..., l_p\}$. Similarly, let $N_2(g^2) = \{k_1, ..., k_q\}$ if $g^2 \notin N_2(g^2)$, otherwise let $N_2(g^2) = \{g^2, k_1, ..., k_q\}$. Then we define $\mathcal{F} : \mathbb{B}^{|N(g^c)|} \to \mathbb{B}$ using four cases as follows:
i) If $g^1 \notin N_1(g^1)$ and $g^2 \notin N_2(g^2)$, then

$$\mathcal{F}(l_1, ..., l_p, k_1, ..., k_q) = F_1(g^1)(l_1, ..., l_p) \odot F_2(g^2)(k_1, ..., k_q);$$

ii) If $g^1 \in N_1(g^1)$ and $g^2 \notin N_2(g^2)$, then

$$\mathcal{F}(g^c, l_1, ..., l_p, k_1, ..., k_q) = F_1(g^1)(g^c, l_1, ..., l_p) \odot F_2(g^2)(k_1, ..., k_q);$$

iii) If $g^1 \notin N_1(g^1)$ and $g^2 \in N_2(g^2)$, then

$$\mathcal{F}(g^c, l_1, ..., l_p, k_1, ..., k_q) = F_1(g^1)(l_1, ..., l_p) \odot F_2(g^2)(g^c, k_1, ..., k_q);$$

iv) If $g^1 \in N_1(g^1)$ and $g^2 \in N_2(g^2)$, then

$$\mathcal{F}(g^c, l_1, ..., l_p, k_1, ..., k_q) = F_1(g^1)(g^c, l_1, ..., l_p) \odot F_2(g^2)(g^c, k_1, ..., k_q).$$

In the sequel, we let g^c denote the new entity created by merging g^1 and g^2 and assume $\mathcal{C}^{\odot}\left(\mathcal{BN}_1, \mathcal{BN}_2, g^1, g^2\right)$ has global states $(g^c\ g^1_1...g^1_n\ g^2_1...g^2_m) \in \mathcal{S}_{\mathcal{C}}$.

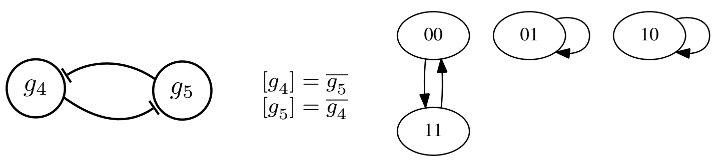

Figure 3: A second Boolean network example $\mathcal{BN}_{Ex2}$ containing the wiring diagram, next–state equations, and state graph.

As an example, consider composing $\mathcal{BN}_{Ex1}$ (Figure 1) and $\mathcal{BN}_{Ex2}$ (Figure 3) on entities g_1 and g_4 using conjunction $\wedge$. The resulting Boolean network $\mathcal{C}^{\wedge}\left(\mathcal{BN}_{Ex1}, \mathcal{BN}_{Ex2}, g_1, g_4\right)$ is depicted in Figure 4.

We would like to be able to infer properties and behaviour of a composed model from the underlying Boolean networks used in the composition. Being able to do this would allow us to construct compositionally large Boolean networks with known properties and so help to address the limitations imposed by the state space explosion problem.

The following definitions formalize the idea that the original behaviour of the underlying Boolean networks can be preserved in their composition. We begin by defining *projection operators* which are able to extract states and traces from a composed system.

Definition 3. (Projections) Let $\mathcal{C} = \mathcal{C}^{\odot}\left(\mathcal{BN}_1, \mathcal{BN}_2, g^1, g^2\right)$ be the new Boolean network constructed by composing $\mathcal{BN}_1$ and $\mathcal{BN}_2$ on entities g^1 and g^2 using $\odot$. Let $S = (g^c\ g^1_1\ ...\ g^1_n\ g^2_1\ ...\ g^2_m) \in \mathcal{S}_{\mathcal{C}}$ be a global state in the composed

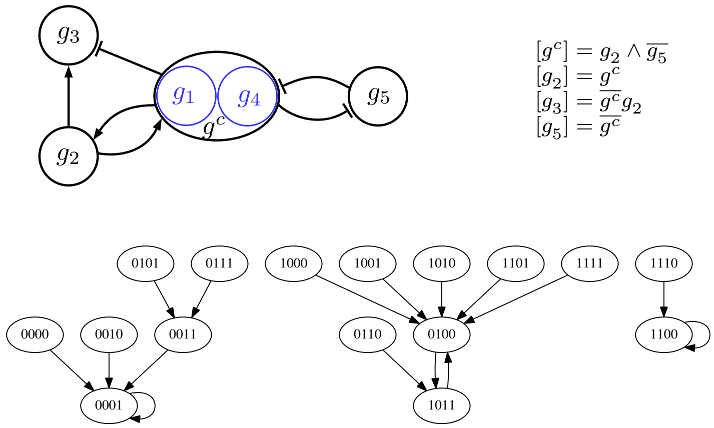

Figure 4: Boolean network $\mathcal{C}^{\wedge}\left(\mathcal{BN}_{Ex1}, \mathcal{BN}_{Ex2}, g_1, g_4\right)$ resulting from the composition of $\mathcal{BN}_{Ex1}$ and $\mathcal{BN}_{Ex2}$ on entities g_1 and g_4 using conjunction $\wedge$ (where the state order is $(g^c\ g_2\ g_3\ g_5)$).

system. Then we define the *left* $\mathcal{P}_1 : \mathcal{S}_{\mathcal{C}} \to \mathcal{S}_{\mathcal{BN}1}$ and *right* $\mathcal{P}_2 : \mathcal{S}_{\mathcal{C}} \to \mathcal{S}_{\mathcal{BN}2}$ *projection operators* by

$$\mathcal{P}_1(S) = (g^c\ g_1^1\ ...\ g_n^1), \quad \mathcal{P}_2(S) = (g^c\ g_1^2\ ...\ g_m^2)$$

We can extend the projection operators to traces $\sigma = \langle S_1, S_2, ...\rangle \in Tr(\mathcal{C})$ by

$$\mathcal{P}_1(\sigma) = \langle \mathcal{P}_1(S_1), \mathcal{P}_1(S_2), ...\rangle, \quad \mathcal{P}_2(\sigma) = \langle \mathcal{P}_2(S_1), \mathcal{P}_2(S_2), ...\rangle$$

and let $\mathcal{P}_1(Tr(\mathcal{C}))$ and $\mathcal{P}_2(Tr(\mathcal{C}))$ represent the sets of projected traces derived by projecting each trace in $Tr(\mathcal{C})$.

Note that projected traces may not be well–defined traces in their corresponding Boolean network, i. e. $\mathcal{P}_j(Tr(\mathcal{C})) \not\subseteq Tr(\mathcal{BN}_j)$ may hold, for $j \in \{1, 2\}$.

We are interested in situations where composing two Boolean networks preserves their underlying trace behaviour and define a notion of *compatibility* to formalise this.

Definition 4. (Compatibility) Let $\mathcal{C} = \mathcal{C}^{\odot}\left(\mathcal{BN}_1, \mathcal{BN}_2, g^1, g^2\right)$ be the Boolean network resulting from composing $\mathcal{BN}_1$ and $\mathcal{BN}_2$ on entities g^1 and g^2 using $\odot$. Then we say that $\mathcal{BN}_1$ and $\mathcal{BN}_2$ are *compatible* on g^1 and g^2 under $\odot$ iff $Tr(\mathcal{BN}_1) \subseteq \mathcal{P}_1(Tr(\mathcal{C}))$ and $Tr(\mathcal{BN}_2) \subseteq \mathcal{P}_2(Tr(\mathcal{C}))$.

To illustrate the definition of compatibility consider composing $\mathcal{BN}_{Ex1}$ and $\mathcal{BN}_{Ex2}$ using conjunction $\wedge$ to produce $\mathcal{C} = \mathcal{C}^{\wedge}(\mathcal{BN}_{Ex1}, \mathcal{BN}_{Ex2}, g_1, g_4)$ (see Figure 4). Then examples of projected traces in $\mathcal{P}_2(Tr(\mathcal{C}))$ (assuming state order $(g^c\ g_2\ g_3\ g_5)$) will be

$\mathcal{P}_2(\langle 0100, 1011, 0100\rangle) = \langle 00, 11, 00\rangle$ $\quad$ $\mathcal{P}_2(\langle 0001, 0001\rangle) = \langle 01, 01\rangle$
$\mathcal{P}_2(\langle 1001, 0100, 1011, 0100\rangle) = \langle 11, 00, 11\rangle$ $\quad$ $\mathcal{P}_2(\langle 1100, 1100\rangle) = \langle 10, 10\rangle$

It can be seen that $Tr(\mathcal{BN}_{Ex1}) \subseteq \mathcal{P}_1(Tr(\mathcal{C}))$ and $Tr(\mathcal{BN}_{Ex2}) \subseteq \mathcal{P}_2(Tr(\mathcal{C}))$ and so we have that $\mathcal{BN}_{Ex1}$ and $\mathcal{BN}_{Ex2}$ are compatible on g_1 and g_4 under $\wedge$.

Compatibility is important since it ensures that properties of the composed model can be inferred from the underlying models used in the composition which are normally much smaller and so simpler to analyse. In particular, if two Boolean networks are compatible then the behaviour they exhibit (such as attractors) must be present in the composed model. Since a composed model may extend the behaviour of its underlying models it does not allow the absence of behaviour to be verified. However, the work presented in Section 5 on interference graphs and in particular, Theorem 5 provides a starting point for addressing these types of negative properties.

Given a Boolean operator $\odot$ which is commutative and associative, we can show that the composition of Boolean networks under $\odot$ is commutative and associative (see [10] for more detail). This result is important as it means that the order in which multiple Boolean networks are composed under a commutative and associative Boolean operator does not affect the resulting model.

4. Compatibility and Alignment

Compatibility is an important behavioural preservation property to consider when composing Boolean networks but unfortunately it is difficult to check since it refers to the full behaviour of the composed model. In this section we begin to address this issue by investigating how to infer compatibility without using the composed model. We formalise a trace alignment property which we show is sufficient for obtaining compatibility in a composed system. We use this result to show that duplicate Boolean networks are compatible under composition of corresponding entities.

For any Boolean network $\mathcal{BN}$ with entities $G = \{g_1, ..., g_n\}$, global state $S = (s_1...s_n) \in S_{\mathcal{BN}}$ and any entity $g_i \in \mathcal{BN}$ we define the state projection $\mathcal{P}_{g_i}(S) = s_i$. Then $\mathcal{P}_{g_i}(\sigma)$ denotes the *projected trace* of entity $g_i \in \mathcal{BN}$ on trace $\sigma = \langle S_1, S_2, ...\rangle \in Tr(\mathcal{BN})$ defined by $\mathcal{P}_{g_i}(\sigma) = \langle \mathcal{P}_{g_i}(S_1), \mathcal{P}_{g_i}(S_2), ...\rangle$. We let $\mathcal{P}_{g_i}(Tr(\mathcal{BN})) = \{\mathcal{P}_{g_i}(\sigma) |\ \sigma \in Tr(\mathcal{BN})\}$. As an example, consider projecting the traces of $\mathcal{BN}_{Ex2}$ (Figure 3) on g_4 which gives $\mathcal{P}_{g_4}(Tr(\mathcal{BN}_{Ex2})) = \{\langle 0,1,0\rangle, \langle 0,0\rangle, \langle 1,1\rangle, \langle 1,0,1\rangle\}$.

We can now define the property of *alignment* as follows.

Definition 5. (Alignment) Let $\mathcal{BN}_1$ and $\mathcal{BN}_2$ be two Boolean networks and let $g^1 \in \mathcal{BN}_1$ and $g^2 \in \mathcal{BN}_2$. We say that $\mathcal{BN}_1$ and $\mathcal{BN}_2$ are *aligned* on g^1 and g^2 iff

$$\mathcal{P}_{g^1}(Tr(\mathcal{BN}_1)) = \mathcal{P}_{g^2}(Tr(\mathcal{BN}_2))$$

In the context of a composition $\mathcal{C} = \mathcal{C}^{\odot}\left(\mathcal{BN}_1, \mathcal{BN}_2, g^1, g^2\right)$ we can define the merging of global states and traces as follows. Let $S^1 = (g^1\ g^1_1\ ...\ g^1_n) \in \mathcal{S}_{\mathcal{BN}1}$ and $S^2 = (g^2\ g^2_1\ ...\ g^2_m) \in \mathcal{S}_{\mathcal{BN}2}$. Then we define $S^1 \odot S^2 \in \mathcal{S}_{\mathcal{C}}$ by merging the state of g^1 with g^2, that is $S^1 \odot S^2 = (g^1 \odot g^2\ g^1_1\ ...\ g^1_n\ g^2_1\ ...\ g^2_m)$. Let $\sigma_1 = \langle S^1_1, S^1_2, ...\rangle \in Tr(\mathcal{BN}_1)$ and $\sigma_2 = \langle S^2_1, S^2_2, ...\rangle \in Tr(\mathcal{BN}_2)$ be two traces.

Then we define $\sigma_1 \odot \sigma_2 = \langle S_1^1 \odot S_1^2, S_2^1 \odot S_2^2, ...\rangle$. Note that for any $\sigma_1 \in Tr(\mathcal{BN}_1)$ and $\sigma_2 \in Tr(\mathcal{BN}_2)$ we may have that $\sigma_1 \odot \sigma_2 \notin Tr(\mathcal{C})$.

Lemma 2. *Let $\mathcal{BN}_1$ and $\mathcal{BN}_2$ be Boolean networks with $G_1 = \{g^1, g_1^1, ..., g_n^1\}$ and $G_2 = \{g^2, g_1^2, ..., g_m^2\}$. Let $\odot$ be an idempotent binary Boolean operator and let $\mathcal{C} = \mathcal{C}^{\odot}\left(\mathcal{BN}_1, \mathcal{BN}_2, g^1, g^2\right)$. Let $\sigma_1 \in Tr(\mathcal{BN}_1)$ and $\sigma_2 \in Tr(\mathcal{BN}_2)$ such that $\mathcal{P}_{g^1}(\sigma_1) = \mathcal{P}_{g^2}(\sigma_2)$. Then we have:*
i) $\sigma_1 \odot \sigma_2 \in Tr(\mathcal{C})$; and
ii) $\mathcal{P}_1(\sigma_1 \odot \sigma_2) = \sigma_1$ and $\mathcal{P}_2(\sigma_1 \odot \sigma_2) = \sigma_2$.

Proof. Let $\sigma_1 = \langle S_1, S_2, ...\rangle \in Tr(\mathcal{BN}_1)$, $\sigma_2 = \langle T_1, T_2, ...\rangle \in Tr(\mathcal{BN}_2)$, and suppose for any i we have $S_i = (s^i\ s_1^i\ ...\ s_n^i)$ and $T_i = (t^i\ t_1^i\ ...\ t_m^i)$. Then by the assumption $\mathcal{P}_{g^1}(\sigma_1) = \mathcal{P}_{g^2}(\sigma_2)$ we know $s^i = t^i$ and so since $\odot$ is assumed to be idempotent we have

$$s^i \odot t^i = s^i = t^i \tag{I}$$

Then the proof for i) and ii) are straightforward based on (I) (see [10] for an example of this proof based on using conjunction). □

We can now prove that *alignment* is a sufficient property for *compatibility.*

Theorem 3. *Let $\mathcal{BN}_1$ and $\mathcal{BN}_2$ be two Boolean networks with $g^1 \in \mathcal{BN}_1$ and $g^2 \in \mathcal{BN}_2$. Let $\odot$ be an idempotent binary Boolean operator. Then if $\mathcal{BN}_1$ and $\mathcal{BN}_2$ are aligned on g^1 and g^2 then $\mathcal{BN}_1$ and $\mathcal{BN}_2$ are compatible on g^1 and g^2 under $\odot$.*

Proof. Let $\mathcal{C} = \mathcal{C}^{\odot}\left(\mathcal{BN}_1, \mathcal{BN}_2, g^1, g^2\right)$ and assume that $\mathcal{BN}_1$ and $\mathcal{BN}_2$ are *aligned* on g^1 and g^2. By the definition of compatibility (Definition 4) we have two properties to show:
i) $Tr(\mathcal{BN}_1) \subseteq \mathcal{P}_1(Tr(\mathcal{C}))$: For any $\sigma_1 \in Tr(\mathcal{BN}_1)$ we know by the alignment assumption above that there exists $\sigma_2 \in Tr(\mathcal{BN}_2)$ such that $\mathcal{P}_{g^1}(\sigma_1) = \mathcal{P}_{g^2}(\sigma_2)$. It follows by Lemma 2.i) that $\sigma_1 \odot \sigma_2 \in Tr(\mathcal{C})$. Then by Lemma 2.ii) we have $\mathcal{P}_1(\sigma_1 \odot \sigma_2) = \sigma_1$ and so $\sigma_1 \in \mathcal{P}_1(Tr(\mathcal{C}))$ as required.
ii) $Tr(\mathcal{BN}_2) \subseteq \mathcal{P}_2(Tr(\mathcal{C}))$: The proof follows along similar lines to i) above. □

The above result provides a means of ensuring compatibility holds without requiring the composed system to be considered. This is important since a composed model will be larger and so more affected by the state space explosion problem. Note that while alignment is a *sufficient* condition for compatibility it can be shown that it is not a *necessary* property for it. We consider this further in the next section where we extend the alignment property to address this.

Two Boolean networks are said to be *duplicates* if they have the same structure and behaviour up to the renaming of entities (i. e. they are isomorphic). It turns out that duplicate Boolean networks are always aligned on corresponding entities (where *corresponding* is defined in the obvious way based on the underlying isomorphism) and therefore are compatible under an idempotent Boolean operator. As an illustration, consider the example presented in Figure 5 based on composing two duplicate copies of $\mathcal{BN}_{Ex1}$ (Figure 1) using conjunction. We

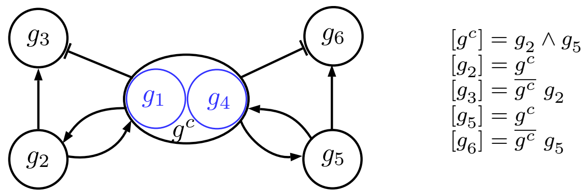

Figure 5: Composing two duplicate copies of $\mathcal{BN}_{Ex1}$ on corresponding entities g_1 and g_4 using conjunction.

can therefore use the alignment property to formally show that duplicate Boolean networks are always compatible when composed on corresponding entities using an idempotent Boolean operator.

Theorem 4. *Let $\mathcal{BN}_1$ and $\mathcal{BN}_2$ be two duplicate Boolean networks and let $g^1 \in \mathcal{BN}_1$ and $g^2 \in \mathcal{BN}_2$ be corresponding entities. Let $\odot$ be an idempotent binary Boolean operator. Then $\mathcal{BN}_1$ and $\mathcal{BN}_2$ are compatible on g^1 and g^2 under $\odot$.*

Proof. Since $\mathcal{BN}_1$ and $\mathcal{BN}_2$ are duplicates it follows (assuming a corresponding state order) that $Tr(\mathcal{BN}_1) = Tr(\mathcal{BN}_2)$. Thus by Definition 5 we know that $\mathcal{BN}_1$ and $\mathcal{BN}_2$ are aligned on corresponding entities g^1 and g^2. Since $\odot$ is assumed to be idempotent we therefore have by Theorem 3 that $\mathcal{BN}_1$ and $\mathcal{BN}_2$ are *compatible* on g^1 and g^2 under $\odot$, as required. $\square$

5. Fully Characterising Compatibility

The alignment property is a *sufficient* property for ensuring compatibility but is limited as it does not completely characterise the compatibility property. To illustrate this consider the counter example $\mathcal{C}^{\wedge}(\mathcal{BN}_{Ex1}, \mathcal{BN}_{Ex3}, g_1, g_4)$ (see Figure 6.(B)), which is the Boolean network resulting from composing $\mathcal{BN}_{Ex1}$ (Figure 1) and $\mathcal{BN}_{Ex3}$ (Figure 6.(A)) on g_1 and g_4 using conjunction. It can be seen that $\mathcal{BN}_{Ex1}$ and $\mathcal{BN}_{Ex3}$ are compatible on g_1 and g_4 under $\wedge$. However, $\mathcal{BN}_{Ex1}$ and $\mathcal{BN}_{Ex3}$ do not align on g_1 and g_4 since there is a trace $\langle 00, 00 \rangle \in Tr(\mathcal{BN}_{Ex1})$ whose projection $\mathcal{P}_{g_1}(\langle 00, 00 \rangle) = \langle 0, 0 \rangle$ is not in $\mathcal{P}_{g_4}(Tr(\mathcal{BN}_{Ex3}))$. Therefore, we have that the alignment property is a *sufficient* but not a *necessary* property for compatibility.

In this section we extend the alignment property to provide a complete characterisation of compatibility. We do this by considering the interference that can occur to the behaviour of Boolean networks when they are composed. We develop a labelled state graph approach to formalise this interference and then use this to define a new alignment property referred to as *weak alignment*. We show that weak alignment completely characterises compatibility (i. e. it is a sufficient and necessary property for compatibility).

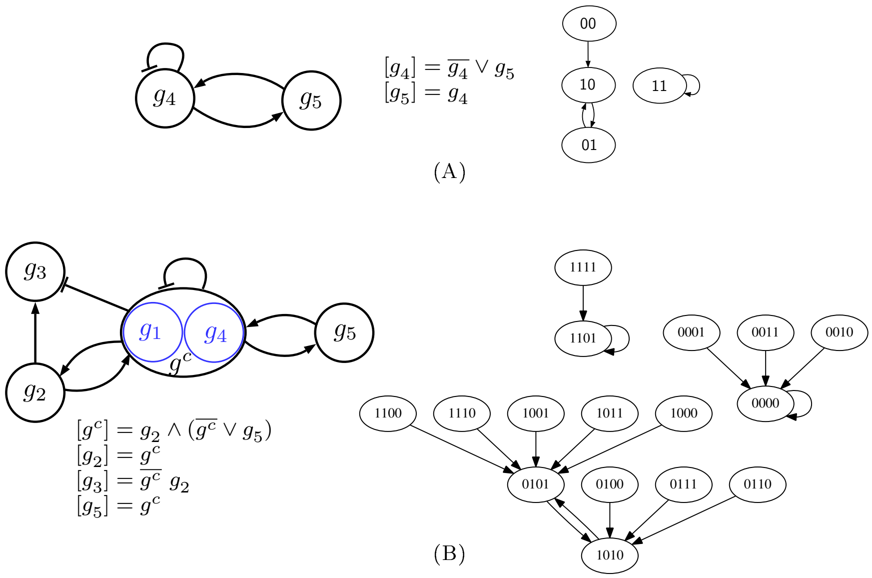

Figure 6: A counter example which shows that alignment is not a necessary property for compatibility. (A) The Boolean network $\mathcal{BN}_{Ex3}$; and (B) Boolean network $\mathcal{C}^{\wedge}\left(\mathcal{BN}_{Ex1}, \mathcal{BN}_{Ex3}, g_1, g_4\right)$ resulting from the composition of $\mathcal{BN}_{Ex1}$ and $\mathcal{BN}_{Ex3}$ on entities g_1 and g_4 using $\wedge$.

5.1. Labelled State Graphs

Recall that $SG(\mathcal{BN}) = (\mathcal{S}_{\mathcal{BN}}, \xrightarrow{\mathcal{BN}})$ is defined to be the state graph for a Boolean network $\mathcal{BN}$ (see Section 2). Let $Path(SG(\mathcal{BN}))$ denote the set of all *infinite paths* in the state graph $SG(\mathcal{BN})$. It is straightforward to see that under the synchronous update semantics traces and infinite paths are equivalent and in the sequel we use these terms interchangeably.

Next we introduce the idea of a *labelled state graph* to allow the projection of a Boolean network's behaviour on a subset of its entities (similar to the approach used in the previous section for traces). Let $\mathcal{BN}$ be a Boolean network and let L be any non-empty set of labels. Then any function $\mathcal{L} : \mathcal{S}_{\mathcal{BN}} \to L$ is termed a state labelling function. Given an infinite path $\alpha = S_1, S_2, ... \in Path(SG(\mathcal{BN}))$ we define $\mathcal{L}(\alpha) = \mathcal{L}(S_1), \mathcal{L}(S_2), ...$ and

$$\mathcal{L}(Path(SG(\mathcal{BN}))) = \{\mathcal{L}(\alpha) \mid \alpha \in Path(SG(\mathcal{BN}))\}$$

We can now define a labelled state graph as follows.

Definition 6. (Labelled State Graph) Let $\mathcal{BN}$ be a Boolean network and let $\mathcal{L} : \mathcal{S}_{\mathcal{BN}} \to L$ be a state labelling function. Then we define the *labelled state graph* $\mathcal{L}(SG(\mathcal{BN}))$ by

$$\mathcal{L}(SG(\mathcal{BN})) = (\mathcal{S}_{\mathcal{BN}}, \xrightarrow{\mathcal{BN}}, \mathcal{L})$$

We define $Path(\mathcal{L}(SG(\mathcal{BN}))) = \mathcal{L}(Path(SG(\mathcal{BN})))$.

Let $\mathcal{C} = \mathcal{C}^{\odot}\left(\mathcal{BN}_1, \mathcal{BN}_2, g^1, g^2\right)$ be the result of composing $\mathcal{BN}_1$ and $\mathcal{BN}_2$ on g^1 and g^2 using $\odot$. Then using the projection functions $\mathcal{P}_1 : \mathcal{S}_{\mathcal{C}} \to \mathcal{S}_{\mathcal{BN}1}$ and $\mathcal{P}_2 : \mathcal{S}_{\mathcal{C}} \to \mathcal{S}_{\mathcal{BN}2}$ defined in Section 3 we can extract the relevant behaviour from $\mathcal{C}$ corresponding to $\mathcal{BN}_1$ and $\mathcal{BN}_2$ by using the labelled state graphs $\mathcal{P}_1(SG(\mathcal{C}))$ and $\mathcal{P}_2(SG(\mathcal{C}))$ respectively. As an example, consider $\mathcal{C} = \mathcal{C}^{\wedge}\left(\mathcal{BN}_{Ex1}, \mathcal{BN}_{Ex2}, g_1, g_4\right)$, the Boolean network resulting from the composition of $\mathcal{BN}_{Ex1}$ and $\mathcal{BN}_{Ex2}$ on entities g_1 and g_4 using conjunction (see Figure 4). Then the labelled state graph $\mathcal{P}_1(SG(\mathcal{C}))$ shown in Figure 7 is the result of projecting from $\mathcal{C}$ the relevant behaviour corresponding to $\mathcal{BN}_{Ex1}$.

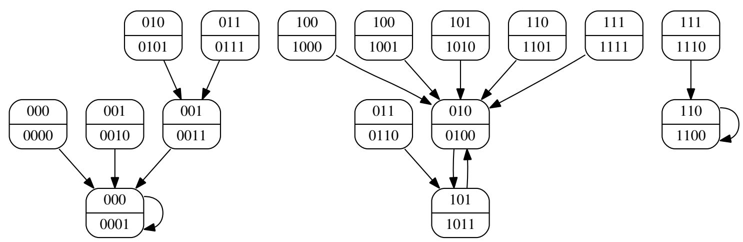

Figure 7: The *labelled state graph* $\mathcal{P}_1(SG(\mathcal{C}))$ derived by applying the left projection $\mathcal{P}_1$ to the state graph $SG(\mathcal{C})$, where $\mathcal{C} = \mathcal{C}^{\wedge}\left(\mathcal{BN}_{Ex1}, \mathcal{BN}_{Ex2}, g_1, g_4\right)$ (Figure 4). Note that each node in $\mathcal{P}_1(SG(\mathcal{C}))$ contains a *bottom part* representing the original global state (g^c g_2 g_3 g_5) and a *top part* representing the projected state (g^c g_2 g_3).

The equivalence between synchronous traces and infinite paths in the state graph means that we can define compatibility and alignment properties in terms of (labelled) state graphs. For example, it is straightforward to show that two Boolean networks $\mathcal{BN}_1$ and $\mathcal{BN}_2$ are *compatible* on g^1 and g^2 under $\odot$ iff we have $Path(SG(\mathcal{BN}_1)) \subseteq Path(\mathcal{P}_1(SG(\mathcal{C})))$ and $Path(SG(\mathcal{BN}_2)) \subseteq Path(\mathcal{P}_2(SG(\mathcal{C})))$, where $\mathcal{C} = \mathcal{C}^{\odot}\left(\mathcal{BN}_1, \mathcal{BN}_2, g^1, g^2\right)$. We also have that $\mathcal{BN}_1$ and $\mathcal{BN}_2$ *align* on g^1 and g^2 iff we have

$$Path(\mathcal{P}_{g^1}(SG(\mathcal{BN}_1))) = Path(\mathcal{P}_{g^2}(SG(\mathcal{BN}_2))).$$

These definitions based on (labelled) state graphs provide a means of developing efficient algorithms for checking the compatibility and alignment properties, and we discuss this further in Section 6.

5.2. Modelling Interference using State Graphs

When two Boolean networks are composed by merging entities it is possible for the behaviour of the underlying models to experience interference resulting in new behaviour. To illustrate this, consider $\mathcal{C}^{\wedge}\left(\mathcal{BN}_{Ex1}, \mathcal{BN}_{Ex2}, g_1, g_4\right)$ (see Figure 4) and suppose the composed model is in global state 0111. Then the underlying Boolean networks will want to make the following transitions:

$011 \xrightarrow{\mathcal{BN}_{Ex1}} 101$ and $01 \xrightarrow{\mathcal{BN}_{Ex2}} 01$. In other words, entity g_1 in $\mathcal{BN}_{Ex1}$ would like to transition from state 0 to state 1 but entity g_4 in $\mathcal{BN}_{Ex2}$ would like to transition 0 to 0. The merged entity g^c will therefore transition to $0 \wedge 1 = 0$ meaning that the underlying behaviour of $\mathcal{BN}_{Ex1}$ has been interfered with. This shows that when using conjunction interference will occur to the underlying behaviour of a merged entity whenever it wants to transition to 1 but its merged counterpart wants to transition to 0. We can formalise this observation by defining an interference set $\Delta = \{1\}$ for conjunction.

For each of the 16 possible different binary Boolean operators we can identify when interference can occur by considering their truth tables and identifying if the value of an argument is preserved after applying the operator. This results in a set of next state values $\Delta \subseteq \mathbb{B}$ which can be interfered with when using a Boolean operator (for example, for conjunction we have $\Delta = \{1\}$ and for disjunction $\Delta = \{0\}$). There are four such possible interference sets $\{\}$, $\{0\}$, $\{1\}$ and $\{0, 1\}$. Note that since not all of the Boolean operators are commutative the order (first or second) of the Boolean network being composed can affect the interference set and for this reason we normally provide two an interference set for the first Boolean network Δ^1 and for the second Δ^2. The above idea is summarised in Table 1 where the interference sets for each possible binary Boolean operator is presented.

A	B	1 F	2 AND	3 AND!	4 A	5 !AND	6 B	7 XOR	8 OR	9 NOR	10 XNOR	11 !B	12 OR!	13 !A	14 !OR	15 NAND	16 T
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
Δ^1		$\{1\}$	$\{1\}$	$\{1\}$	$\{\}$	$\{0, 1\}$	$\{0, 1\}$	$\{0, 1\}$	$\{0\}$	$\{0, 1\}$	$\{0, 1\}$	$\{0, 1\}$	$\{0\}$	$\{0, 1\}$	$\{0, 1\}$	$\{0, 1\}$	$\{0\}$
Δ^2		$\{1\}$	$\{1\}$	$\{0, 1\}$	$\{0, 1\}$	$\{1\}$	$\{\}$	$\{0, 1\}$	$\{0\}$	$\{0, 1\}$	$\{0, 1\}$	$\{0, 1\}$	$\{0, 1\}$	$\{0, 1\}$	$\{0\}$	$\{0, 1\}$	$\{0\}$

Table 1: The interference sets Δ^1 (first Boolean network) and Δ^2 (second Boolean network) for each of the 16 binary Boolean operators.

In order to formalise the possible *interference behaviour* that can occur between composed Boolean networks we extend a Boolean network's state graph with additional edges representing interference. The idea is that whenever the entity to be merged transitions to a state s that can experience interference $s \in \Delta$ then we add another edge to the state graph to represent that the transition could instead go to $\neg s$. This situation is depicted in Figure 8 based on the interference example presented above for $\mathcal{BN}_{Ex1}$ using conjunction.

We formalise this idea by defining an *interference state graph* as follows.

Definition 7. (Interference State Graph) Let $\mathcal{BN}$ be a Boolean network with entities $g, g_1, ..., g_n$ and let $\Delta \subseteq \mathbb{B}$ be an interference set. Then we define the

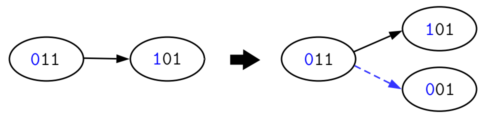

Figure 8: Example based on $\mathcal{BN}_{Ex1}$ (Figure 1) which illustrates how interference $\Delta = \{1\}$ (conjunction) can be modelled by adding additional edges (dashed edge) to a Boolean network's state graph.

interference state graph $SG_{g,\Delta}(\mathcal{BN})$ for $\mathcal{BN}$ on g with interference Δ by

$$SG_{g,\Delta}(\mathcal{BN}) = (\mathcal{S}_{\mathcal{BN}}, \xrightarrow[g,\Delta]{\mathcal{BN}})$$

The extended edge relation $\xrightarrow[g,\Delta]{\mathcal{BN}}$ is defined by $\xrightarrow[g,\Delta]{\mathcal{BN}} = \xrightarrow{\mathcal{BN}} \cup \mathcal{E}$ where

$$\mathcal{E} = \{((s\ s_1\ ...\ s_n), (\neg s'\ s'_1\ ...\ s'_n)) \mid (s\ s_1\ ...\ s_n) \xrightarrow{\mathcal{BN}} (s'\ s'_1\ ...\ s'_n),\ s' \in \Delta\}$$

To illustrate this definition, consider the interference state graph $SG_{g_1,\Delta}(\mathcal{BN}_{Ex_1})$ depicted in Figure 9, where four new edges (given as dashed arrows) are needed to capture the interference $\Delta = \{1\}$ that can occur to the behaviour of $\mathcal{BN}_{Ex1}$ during a composition under conjunction.

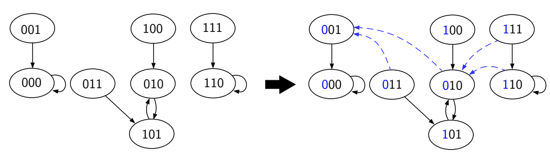

Figure 9: Example illustrating how the interference state graph $SG_{g_1,\Delta}(\mathcal{BN}_{Ex1})$ is derived from the state graph $SG(\mathcal{BN}_{Ex1})$ by adding edges (dashed arrows) to model the interference on g_1 that can occur during composition using conjunction ($\Delta = \{1\}$).

Interestingly, it turns out that the interference state graph is able to capture all the possible behaviours a Boolean network can have under composition. As an example, consider Figure 10 which shows that all paths contained within the projected state graph $\mathcal{P}_1(SG(\mathcal{C}^{\wedge}(\mathcal{BN}_{Ex1}, \mathcal{BN}_{Ex2}, g_1, g_4)))$ are also contained in the interference state graph $SG_{g_1,\Delta}(\mathcal{BN}_{Ex_1})$. This observation is formally proved in the following theorem.

Theorem 5. *Let $\mathcal{BN}_1$ and $\mathcal{BN}_2$ be two Boolean networks, and let $g^1 \in \mathcal{BN}_1$, $g^2 \in \mathcal{BN}_2$. Let $\mathcal{C} = \mathcal{C}^{\odot}\left(\mathcal{BN}_1, \mathcal{BN}_2, g^1, g^2\right)$, and let Δ^1 and Δ^2 be the first and second interference sets for $\odot$. Then we have:*
i) $Path(\mathcal{P}_1(SG(\mathcal{C}))) \subseteq Path(SG_{g^1,\Delta^1}(\mathcal{BN}_1))$; and
ii) $Path(\mathcal{P}_2(SG(\mathcal{C}))) \subseteq Path(SG_{g^2,\Delta^2}(\mathcal{BN}_2))$.

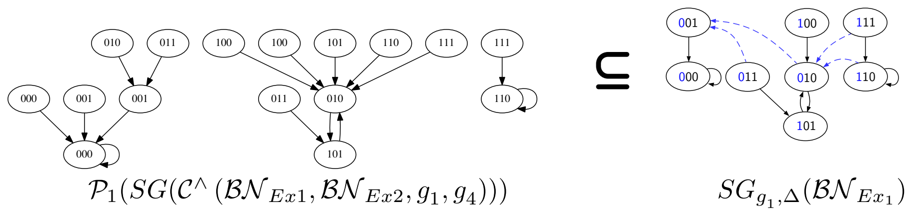

Figure 10: Example illustrating that the interference state graph $SG_{g_1,\Delta}(\mathcal{BN}_{Ex_1})$ contains all the behaviour in the projected state graph $\mathcal{P}_1(SG(\mathcal{C}^\wedge(\mathcal{BN}_{Ex1}, \mathcal{BN}_{Ex2}, g_1, g_4)))$ (where $\Delta = \{1\}$ represents the interference associated with conjunction).

Proof. i) Let $\beta = S_1, S_2, ... \in Path(SG(\mathcal{C}))$. Then we need to show there exists a path $\alpha \in Path(SG_{g^1,\Delta^1}(\mathcal{BN}_1))$ such that $\mathcal{P}_1(\beta) = \alpha$. It suffices to show that for any $i \in \mathbb{N}$ there exists an edge

$$\mathcal{P}_1(S_i) \xrightarrow[g^1,\Delta^1]{\mathcal{BN}_1} \mathcal{P}_1(S_{i+1})$$

in the interference graph $SG_{g^1,\Delta^1}(\mathcal{BN}_1)$. Let $S_i = (st\ s_1^i\ ...\ s_n^i\ t_1^i\ ...\ t_m^i) \in \mathcal{S}_\mathcal{C}$, where $st \in \mathbb{B}$ is the state of the merged entity $g^1 \odot g^2$. By definition of projection we know

$$\mathcal{P}_1(S_i) = (st\ s_1^i\ ...\ s_n^i), \text{ and } \mathcal{P}_2(S_i) = (st\ t_1^i\ ...\ t_m^i)$$

Suppose that

$$(st\ s_1^i\ ...\ s_n^i) \xrightarrow{\mathcal{BN}_1} (s^{i+1}\ s_1^{i+1}\ ...\ s_n^{i+1}) \tag{I}$$

$$(st\ t_1^i\ ...\ t_m^i) \xrightarrow{\mathcal{BN}_2} (t^{i+1}\ t_1^{i+1}\ ...\ t_m^{i+1}) \tag{II}$$

Then it follows by Definition 2, (I) and (II) that

$$S_{i+1} = (s^{i+1} \odot t^{i+1}\ s_1^{i+1}\ ...\ s_n^{i+1}\ t_1^{i+1}\ ...\ t_m^{i+1})$$

We now have two cases to consider with respect to $s^{i+1} \odot t^{i+1}$:
1) Suppose $s^{i+1} \odot t^{i+1} = s^{i+1}$ (i.e. there was no interference for $\mathcal{BN}_1$). Then by the definition of projection we have

$$\mathcal{P}_1(S_{i+1}) = (s^{i+1}\ s_1^{i+1}\ ...\ s_n^{i+1})$$

Then by (I) we know that

$$\mathcal{P}_1(S_i) \xrightarrow{\mathcal{BN}_1} \mathcal{P}_1(S_{i+1})$$

and so the result follows since by Definition 7 we know all the edges in $SG(\mathcal{BN}_1)$ are contained in $SG_{g^1,\Delta^1}(\mathcal{BN}_1)$.
2) Suppose $s^{i+1} \odot t^{i+1} \neq s^{i+1}$ (i.e. there was interference for $\mathcal{BN}_1$). Then by the definition of projection we must have

$$\mathcal{P}_1(S_{i+1}) = (\neg s^{i+1}\ s_1^{i+1}\ ...\ s_n^{i+1})$$

By the definition of the interference set Δ^1 this case can happen only when $s^{i+1} \in \Delta^1$ and so by Definition 7 and (I), we must have added the edge

$$(st\ s_1^i\ ...\ s_n^i) \xrightarrow[g^1,\Delta^1]{\mathcal{BN}_1} (\neg s^{i+1}\ s_1^{i+1}\ ...\ s_n^{i+1})$$

to $SG_{g^1,\Delta^1}(\mathcal{BN}_1)$ as required.
ii) Follows along similar lines to i) above. □

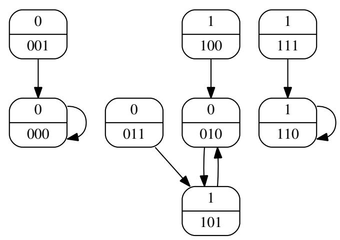

Figure 11: A labelled state graph $\mathcal{P}_{g_1}(SG(\mathcal{BN}_{Ex1}))$ based on projecting the state of a single entity g_1 (where the entity order in the global state is $(g_1\ g_2\ g_3)$).

5.3. Weak Alignment

In this section we use the interference state graph to extend the alignment property in order to fully characterise compatibility. The resulting property is referred to as *weak alignment* and we formally show that it is both sufficient and necessary for compatibility.

Recall the entity state projection function $\mathcal{P}_{g_i} : S_{\mathcal{BN}} \to \mathbb{B}$ defined for any $(s_1...s_n) \in S_{\mathcal{BN}}$ by $\mathcal{P}_{g_i}(s_1...s_n) = s_i$ (see Section 4). We can use this projection to extract from a state graph the behaviour of a single entity. As an example, consider Figure 11 which depicts the labelled state graph $\mathcal{P}_{g_1}(SG(\mathcal{BN}_{Ex1}))$ representing the behaviour of entity g_1 in $\mathcal{BN}_{Ex1}$.

We formally define *weak alignment* using the interference graph as follows.

Definition 8. (Weak Alignment) Let $\mathcal{BN}_1$ and $\mathcal{BN}_2$ be Boolean networks, and let $g^1 \in \mathcal{BN}_1$ and $g^2 \in \mathcal{BN}_2$. Let $\odot$ be a binary Boolean operator, and let Δ^1 and Δ^2 be the first and second interference sets for $\odot$. Then we say that $\mathcal{BN}_1$ and $\mathcal{BN}_2$ are *weakly aligned* on g^1 and g^2 under $\odot$ iff

$$Path(\mathcal{P}_{g^1}(SG(\mathcal{BN}_1))) \subseteq Path(\mathcal{P}_{g_2}(SG_{g^2,\Delta^2}(\mathcal{BN}_2))),\ \text{and}$$

$$Path(\mathcal{P}_{g^2}(SG(\mathcal{BN}_2))) \subseteq Path(\mathcal{P}_{g_1}(SG_{g^1,\Delta^1}(\mathcal{BN}_1)))$$

Weak alignment is a more expressive property than alignment and is able to capture the fundamental relationship that exists between compatible Boolean networks. To illustrate this, consider again the counter example presented in Figure 6. In this example we have $\mathcal{P}_{g_4}(Tr(\mathcal{BN}_{Ex3})) \subseteq \mathcal{P}_{g_1}(Tr(\mathcal{BN}_{Ex1}))$ but

$\mathcal{P}_{g_1}(Tr(\mathcal{BN}_{Ex1})) \not\subseteq \mathcal{P}_{g_4}(Tr(\mathcal{BN}_{Ex3}))$ and so $\mathcal{BN}_{Ex1}$ and $\mathcal{BN}_{Ex3}$ do not align on g_1 and g_4. However, as Figure 12 shows we do have

$$Path(\mathcal{P}_{g_1}(SG(\mathcal{BN}_{Ex1}))) \subseteq Path(\mathcal{P}_{g_4}(SG_{g_4,\Delta}(\mathcal{BN}_{Ex3})))$$

(where $\Delta = \{1\}$ for conjunction) since interference on $\mathcal{BN}_{Ex3}$ provides the missing path $00, 00, \ldots$. It therefore follows that $\mathcal{BN}_{Ex1}$ and $\mathcal{BN}_{Ex3}$ weakly align on g_1 and g_4 under $\wedge$.

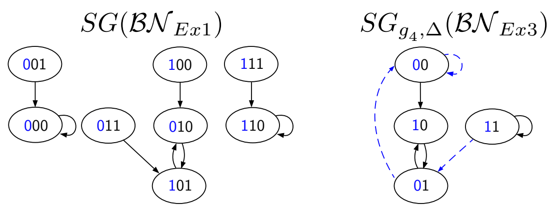

Figure 12: State graph for $\mathcal{BN}_{Ex1}$ and interference state graph for $\mathcal{BN}_{Ex3}$ (where $\Delta = \{1\}$ for conjunction) which show that $Path(\mathcal{P}_{g_1}(SG(\mathcal{BN}_{Ex1}))) \subseteq Path(SG_{g_4,\Delta}(\mathcal{BN}_{Ex3}))$ (where the global states have entity order $(g_1\ g_2\ g_3)$ and $(g_4\ g_5)$).

Recall from Section 4 that in the context of a composition $\mathcal{C}^{\odot}\left(\mathcal{BN}_1, \mathcal{BN}_2, g^1, g^2\right)$ we can merge traces $\alpha_1 \in Tr(\mathcal{BN}_1)$ and $\alpha_2 \in Tr(\mathcal{BN}_2)$ on g^1 and g^2 using the given binary Boolean operator $\alpha_1 \odot \alpha_2$. Then this idea can be applied to paths in the obvious way.

The following are some useful results that can be shown about merging paths that exhibit the weak alignment property.

Lemma 6. *Let $\mathcal{BN}_1$ and $\mathcal{BN}_2$ be two Boolean networks such that $g^1 \in \mathcal{BN}_1$ and $g^2 \in \mathcal{BN}_2$. Let $\odot$ be a binary Boolean operator which is idempotent and let Δ^1 and Δ^2 be its associated interference sets. Let $\alpha_1 \in Path(SG(\mathcal{BN}_1))$, $\beta_1 \in Path(SG_{g^1,\Delta^1}(\mathcal{BN}_1))$, $\alpha_2 \in Path(SG(\mathcal{BN}_2))$, and $\beta_2 \in Path(SG_{g^2,\Delta^2}(\mathcal{BN}_2))$ be infinite paths such that*

$$\mathcal{P}_{g^1}(\alpha_1) = \mathcal{P}_{g^2}(\beta_2) \quad and \quad \mathcal{P}_{g^2}(\alpha_2) = \mathcal{P}_{g^1}(\beta_1)$$

Then we have:
i) $\alpha_1 \odot \beta_2 \in Path(SG(\mathcal{C}^{\odot}\left(\mathcal{BN}_1, \mathcal{BN}_2, g^1, g^2\right)))$;
ii) $\beta_1 \odot \alpha_2 \in Path(SG(\mathcal{C}^{\odot}\left(\mathcal{BN}_1, \mathcal{BN}_2, g^1, g^2\right)))$;
iii) $\mathcal{P}_1(\alpha_1 \odot \beta_2) = \alpha_1$ and $\mathcal{P}_2(\alpha_1 \odot \beta_2) = \beta_2$; and
iv) $\mathcal{P}_1(\beta_1 \odot \alpha_2) = \beta_1$ and $\mathcal{P}_2(\beta_1 \odot \alpha_2) = \alpha_2$.

PROOF. i) Let $\alpha_1 = S_1, S_2, \ldots \in Path(SG(\mathcal{BN}_1))$ be an infinite path, and let $\beta_2 = T_1, T_2, \ldots \in Path(SG_{g^2,\Delta^2}(\mathcal{BN}_2))$ such that $\mathcal{P}_{g^1}(\alpha_1) = \mathcal{P}_{g^2}(\beta_2)$. Let $\mathcal{C} = \mathcal{C}^{\odot}\left(\mathcal{BN}_1, \mathcal{BN}_2, g^1, g^2\right)$. In order to show that $\alpha_1 \odot \beta_2 \in Path(SG(\mathcal{C}))$ it suffices to show that for each $i \in \mathbb{N}$,

$$S_i \odot T_i \xrightarrow{\mathcal{C}} S_{i+1} \odot T_{i+1} \qquad \text{(I)}$$

We know by assumption that for any $i \in \mathbb{N}$,

$$S_i \xrightarrow{\mathcal{BN}_1} S_{i+1} \quad \text{and} \quad T_i \xrightarrow[g^2,\Delta^2]{\mathcal{BN}_2} T_{i+1}$$

Let $S_i = (s^i\ s^i_1\ ...\ s^i_n)$, $S_{i+1} = (s^{i+1}\ s^{i+1}_1\ ...\ s^{i+1}_n)$, $T_i = (t^i\ t^i_1\ ...\ t^i_m)$, and $T_{i+1} = (t^{i+1}\ t^{i+1}_1\ ...\ t^{i+1}_m)$. Then by definition of merging states we have

$$S_i \odot T_i = (s^i \odot t^i\ s^i_1\ ...\ s^i_n\ t^i_1\ ...\ t^i_m)$$

$$S_{i+1} \odot T_{i+1} = (s^{i+1} \odot t^{i+1}\ s^{i+1}_1\ ...\ s^{i+1}_n\ t^{i+1}_1\ ...\ t^{i+1}_m)$$

By assumption $\mathcal{P}_{g^1}(\alpha_1) = \mathcal{P}_{g^2}(\beta_2)$ we know that $s^i = t^i$ and so by idempotence of $\odot$ we have

$$s^i \odot t^i = s^i = t^i$$

Now suppose

$$T_i \xrightarrow{\mathcal{BN}_2} (u^{i+1}, t^{i+1}_1, ..., t^{i+1}_m)$$

Then we know by definition of $\mathcal{C}$ that

$$(s^i \odot t^i\ s^i_1\ ...\ s^i_n\ t^i_1\ ...\ t^i_m) \xrightarrow{\mathcal{C}} (s^{i+1} \odot u^{i+1}\ s^{i+1}_1\ ...\ s^{i+1}_n\ t^{i+1}_1\ ...\ t^{i+1}_m)$$

Therefore in order to prove (I) we have to show that $s^{i+1} \odot t^{i+1} = s^{i+1} \odot u^{i+1}$ which we do by considering the following two cases.
Case 1: Suppose $t^{i+1} = u^{i+1}$ (i.e. there was no interference for $\mathcal{BN}_2$). Then we must have

$$s^{i+1} \odot t^{i+1} = s^{i+1} \odot u^{i+1}$$

as required.
Case 2: Suppose $t^{i+1} \neq u^{i+1}$ (i.e. there was interference for $\mathcal{BN}_2$). It follows by Definition 7 that $u^{i+1} \in \Delta^2$ and $t^{i+1} = \neg u^{i+1}$. Furthermore, since $\odot$ is idempotent we must have $\neg u^{i+1} \odot u^{i+1} = \neg u^{i+1}$ (i.e. $\neg u^{i+1}$ is the value that causes the interference). By assumption $\mathcal{P}_{g^1}(\alpha_1) = \mathcal{P}_{g^2}(\beta_2)$ we know that $s^{i+1} = t^{i+1}$ and so since $\odot$ is idempotent we have

$$s^{i+1} \odot t^{i+1} = s^{i+1}$$

Given that we can infer $s^{i+1} = \neg u^{i+1}$ it follows that

$$s^{i+1} = s^{i+1} \odot u^{i+1}$$

and so $s^{i+1} \odot t^{i+1} = s^{i+1} \odot u^{i+1}$ as required.
ii) Follows along similar lines to i) above and so for brevity is omitted.
iii) We need to show that $\mathcal{P}_1(\alpha_1 \odot \beta_2) = \alpha_1$ and $\mathcal{P}_2(\alpha_1 \odot \beta_2) = \beta_2$. Let $\alpha_1 = S_1, S_2, ... \in Path(SG(\mathcal{BN}_1))$, $\beta_2 = T_1, T_2, ... \in Path(SG_{g2,\Delta^2}(\mathcal{BN}_2))$. Then we have

$$\alpha_1 \odot \beta_2 = S_1 \odot T_1, S_2 \odot T_2, ...$$

For each $i \in \mathbb{N}$, let $S_i = (s^i \ s^i_1 \ ... \ s^i_n)$, and $T_i = (t^i \ t^i_1 \ ... \ t^i_m)$. Since we know $\mathcal{P}_{g^1}(\alpha_1) = \mathcal{P}_{g^2}(\beta_2)$, it follows that $s^i = t^i$ and so since $\odot$ is idempotent we have

$$s^i \odot t^i = s^i = t^i$$

Then it follows that $\mathcal{P}_1(S_i \odot T_i) = S_i$ and $\mathcal{P}_2(S_i \odot T_i) = T_i$ as required.
iv) Follows along similar lines to iii) above and so again is omitted for brevity.
$\square$

We can now formally prove that weak alignment is able to fully characterise compatibility (i. e. weak alignment is a sufficient and necessary property for compatibility).

Theorem 7. *Let $\mathcal{BN}_1$ and $\mathcal{BN}_2$ be two Boolean networks such that $g^1 \in \mathcal{BN}_1$ and $g^2 \in \mathcal{BN}_2$, and let $\odot$ be a binary Boolean operator which is idempotent. Then $\mathcal{BN}_1$ and $\mathcal{BN}_2$ are compatible on g^1 and g^2 under $\odot$ iff $\mathcal{BN}_1$ and $\mathcal{BN}_2$ are weakly aligned on g^1 and g^2 under $\odot$.*

Proof. Let $\mathcal{C} = \mathcal{C}^{\odot}\left(\mathcal{BN}_1, \mathcal{BN}_2, g^1, g^2\right)$ and let Δ^1 and Δ^2 be the interference sets associated with $\odot$. We have two cases to consider.
Case 1: Suppose $\mathcal{BN}_1$ and $\mathcal{BN}_2$ are *compatible* on g^1 and g^2 under $\odot$. Then we need to prove that $\mathcal{BN}_1$ and $\mathcal{BN}_2$ are *weakly aligned* on g^1 and g^2 under $\odot$. By the definition of weak alignment (Definition 8) we need to show

$$Path(\mathcal{P}_{g^1}(SG(\mathcal{BN}_1))) \subseteq Path(\mathcal{P}_{g_2}(SG_{g^2,\Delta^2}(\mathcal{BN}_2))) \quad (1)$$
$$Path(\mathcal{P}_{g^2}(SG(\mathcal{BN}_2))) \subseteq Path(\mathcal{P}_{g_1}(SG_{g^1,\Delta^1}(\mathcal{BN}_1))) \quad (2)$$

To show (1) consider any $\alpha \in Path(SG(\mathcal{BN}_1))$. Then by the assumption of compatibility we know

$$Path(SG(\mathcal{BN}_1)) \subseteq Path(\mathcal{P}_1(SG(\mathcal{C})))$$

and so there must exist an infinite path $\beta \in Path(SG(\mathcal{C}))$ such that $\alpha = \mathcal{P}_1(\beta)$. By Theorem 5 we know that

$$\mathcal{P}_2(\beta) \in Path(SG_{g^2,\Delta^2}(\mathcal{BN}_2))$$

Since we must have $\mathcal{P}_{g^1}(\mathcal{P}_1(\beta)) = \mathcal{P}_{g^2}(\mathcal{P}_2(\beta))$ it follows that

$$\mathcal{P}_{g^1}(\alpha) = \mathcal{P}_{g^2}(\mathcal{P}_2(\beta))$$

Therefore (1) must hold as required.

The proof of (2) follows along similar lines to (1) above and so for brevity is omitted.

Case 2: Suppose $\mathcal{BN}_1$ and $\mathcal{BN}_2$ are *weakly aligned* on g^1 and g^2 under $\odot$. Then we need to prove that $\mathcal{BN}_1$ and $\mathcal{BN}_2$ are *compatible* on g^1 and g^2 under $\odot$. By the definition of compatibility (Definition 4) we need to show

$$Path(SG(\mathcal{BN}_1)) \subseteq Path(\mathcal{P}_1(SG(\mathcal{C}))) \quad (3)$$
$$Path(SG(\mathcal{BN}_2)) \subseteq Path(\mathcal{P}_2(SG(\mathcal{C}))) \quad (4)$$

To show (3) consider any path $\alpha_1 \in Path(SG(\mathcal{BN}_1))$. Then by assumption and the definition of weak alignment (Definition 8) there must exist a path $\alpha_2 \in Path(SG_{g^2,\Delta^2}(\mathcal{BN}_2))$ such that

$$\mathcal{P}_{g^1}(\alpha_1) = \mathcal{P}_{g^2}(\alpha_2)$$

It follows by Lemma 6.i) that

$$\alpha_1 \odot \alpha_2 \in Path(SG(\mathcal{C}))$$

Since by Lemma 6.ii) we know $\mathcal{P}_1(\alpha_1 \odot \alpha_2) = \alpha_1$ the result follows.

The proof of (4) follows along similar lines to (3) above and so again is omitted for brevity. □

6. Conclusions

In this paper we set out to develop a compositional framework for Boolean networks in order to facilitate the construction and analysis of large scale models. This work was motivated by interesting interactions with the synthetic biology group at Newcastle[1] and their search for formal tools and techniques to support their work on engineering biological systems. We have formally defined our compositional approach based on the idea of merging entities between models using binary Boolean operators. In order to formalise the preservation of a Boolean network's behaviour within a composed model we introduced the notion of compatibility. We formulated the alignment property which we showed was a sufficient condition for ensuring compatibility given an idempotent Boolean operator and used it to investigate the composition of duplicate models.

To completely characterise compatibility, we considered formalising the interference that can occur to the behaviour of a Boolean network in a composition using a state graph approach. We used the resulting interference state graph to define the weak alignment property which we then showed completely characterised compatibility given an idempotent Boolean operator (i. e. it is sufficient and necessary for compatibility). This final result is very important since weak alignment allows us to check behaviour preservation (compatibility) without referencing the composed model and so helps avoid potentially limiting state space explosion issues. The compositional framework presented here is supported by a prototype tool that automates the composition process and associated behaviour preservation analysis.

6.1. Related Work

The main focus of existing work on compositional techniques in Boolean networks is on the efficient identification of attractors. The composition of random Boolean networks is considered in [17] and an approach based on identifying

[1] www.ncl.ac.uk/csbb/

independent components of a Boolean network whose composed behaviour can infer attractors is developed. In [18] a compositional approach for asynchronous Boolean networks is considered based on using a state graph approach. Various work has also considered extending the applicability of SAT based techniques by partitioning large models and then aggregating the results (see for example [19, 20]). An algorithmic approach for efficient attractor identification in large-scale Boolean networks based on analysing subnetworks in a model is considered in [21, 22].

An interesting range of research focusing on subnetwork embeddings and asynchronous behaviour preservation in *Logical Regulatory Graphs* [23] can be found in the literature. In [24] the identification of subnetworks that control the overall Boolean asynchronous behaviour of a model has been considered. By extending existing concepts of *singular states* and *local interaction graphs* they are able to identify the key structures influencing the behaviour of the model and so use this to predict its behaviour. The preservation of behaviour is also considered in [25] where they use the parameters associated with the output edges leaving an embedded network to formulate a necessary and sufficient condition for behaviour preservation. Further interesting work on the embedding of networks and modularity in this setting can be found in [26, 27].

In very recent work, a modular approach for *Boolean Automata Networks (BANs)* is presented in [28] where external inputs are added to the base model along with a notion of wiring. The initial results they present indicate a promising approach to investigating the composition and decomposition of BANs.

The approach taken in this paper based on composing synchronous Boolean networks by using binary Boolean operators to merge entities and then characterising behavioural preservation using interference appears to differ significantly from previously published work.

6.2. Future Work

The formal compositional framework developed in this paper provides the basis for a large range of future work. For example, we are currently working to generalise the compositional framework to allow multiple entities and multiple Boolean networks to be composed simultaneously. We have also begun to extend our compositional framework to *multi-valued networks* (see for example [29, 15, 14]), an extension of the Boolean network approach in which an entity's state is allowed to be a range of discrete values instead of just Boolean. In turn we aim to further strengthen the tool support with these results to ensure the practical application of the techniques developed is fully supported. Finally, we are interested in using our compositional framework as the basis for developing formal techniques for *decomposing* Boolean networks. The idea is to develop practical techniques and tools to allow the automatic decomposition of realistically large Boolean networks and so facilitate their analysis and investigation.

Acknowledgments

We would like to thank Will Peckham for his work on developing tool support for our compositional framework. We also acknowledge the financial support provided by Faculty of Computing and Information Technology, King Abdulaziz University. Finally, we would like to thank the anonymous referees for their very helpful comments and suggestions.

References

[1] H. De Jong, Modeling and simulation of genetic regulatory systems: a literature review, Journal of Computational Biology 9 (1) (2002) 67–103.

[2] E. Bartocci, P. Lió, Computational modeling, formal analysis, and tools for systems biology, PLoS Computational Biology 12 (1) (2016) e1004591.

[3] S. A. Kauffman, Metabolic stability and epigenesis in randomly constructed genetic nets, Journal of Theoretical Biology 22 (3) (1969) 437–467.

[4] S. A. Kauffman, The origins of order: Self organization and selection in evolution, Oxford University Press, USA, 1993.

[5] T. Akutsu, S. Miyano, S. Kuhara, et al., Identification of genetic networks from a small number of gene expression patterns under the Boolean network model, Pacific Symposium on Biocomputing 4 (1999) 17–28.

[6] L. J. Steggles, R. Banks, O. Shaw, A. Wipat, Qualitatively modelling and analysing genetic regulatory networks: a petri net approach, Bioinformatics 23 (3) (2007) 336–343.

[7] A. Saadatpour, R. Albert, Boolean modeling of biological regulatory networks: a methodology tutorial, Methods 62 (1) (2013) 3–12.

[8] D. A. Rosenblueth, S. Muñoz, M. Carrillo, E. Azpeitia, Inference of boolean networks from gene interaction graphs using a sat solver, in: A.-H. Dediu, C. Martín-Vide, B. Truthe (Eds.), International Conference on Algorithms for Computational Biology, AlCoB 2014, LNCS 8542, Springer, 2014, pp. 235–246.

[9] S. Kurgalin, S. Borunov, The Discrete Math Workbook: A Companion Manual for Practical Study., Texts in Computer Science, Springer, 2018.

[10] H. Alkhudhayr, J. Steggles, A formal framework for composing qualitative models of biological systems, in: C. MartÃŋn-Vide, R. Neruda, M. Vega-RodrÃŋguez (Eds.), Theory and Practice of Natural Computing, TPNC2017, LNCS 10687, Springer, 2017, pp. 25–36.

[11] A. Wuensche, Aggregation algorithm towards large-scale Boolean network analysis, in: G. Schlosser, G. P. Wagner (Eds.), Modularity in Development and Evolution, Uni Of Chicago Press, 2004, Ch. 13, pp. 288–311.

[12] I. Harvey, T. Bossomaier, Time out of joint: Attractors in asynchronous random Boolean networks, in: Proceedings of the Fourth European Conference on Artificial Life, MIT Press, Cambridge, 1997, pp. 67–75.

[13] M. A. Schaub, T. A. Henzinger, J. Fisher, Qualitative networks: a symbolic approach to analyze biological signaling networks, BMC Systems Biology 1 (4).

[14] R. Banks, L. J. Steggles, An abstraction theory for qualitative models of biological systems, Theoretical Computer Science 431 (2012) 207–218.

[15] D. Thieffry, R. Thomas, Dynamical behaviour of biological regulatory networks—II. Immunity control in bacteriophage lambda, Bulletin of Mathematical Biology 57 (2) (1995) 277–297.

[16] S. Huang, D. E. Ingber, Shape-dependent control of cell growth, differentiation, and apoptosis: switching between attractors in cell regulatory networks, Experimental cell research 261 (1) (2000) 91–103.

[17] E. Dubrova, M. Teslenko, Compositional properties of random Boolean networks, Physical Review E 71 (2005) 056116. doi:10.1103/PhysRevE.71.056116. URL http://link.aps.org/doi/10.1103/PhysRevE.71.056116

[18] L. Tournier, M. Chaves, Interconnection of asynchronous Boolean networks, asymptotic and transient dynamics, Automatica 49 (4) (2013) 884–893.

[19] W. Guo, G. Yang, W. Wu, L. He, M. Sun, A parallel attractor-finding algorithm based on Boolean satisfiability for genetic regulatory networks, PLoS ONE 9 (4) (2014) e94258. doi:10.1371/journal.pone.0094258.

[20] C. Hong, J. Hwang, K.-H. Cho, I. Shin, An efficient steady-state analysis method for large Boolean networks with high maximum node connectivity, PLoS ONE 10 (12) (2015) e0145734. doi:10.1371/journal.pone.0145734.

[21] Y. Zhao, J. Kim, M. Filippone, Compositional properties of random Boolean networks, IEEE Transactions on Automatic Control 58 (2013) 1976–1985.

[22] D. Cheng, Y. Zhao, J. Kim, Y. Zhao, Approximation of Boolean networks, in: Proc. of the 10th World Congress on Intelligent Control and Automation (WCICA), 2012, pp. 2280–2285. doi:10.1109/WCICA.2012.6358254.

[23] R. Thomas, R. d'Ari, Biological Feedback, CRC Press, Boca Raton, 1990.

[24] H. Siebert, Deriving behavior of Boolean bioregulatory networks from subnetwork dynamics, Mathematics in Computer Science 2 (2009) 421–442. doi:10.1007/s11786-008-0064-4.

[25] G. Bernot, F. Tahi, Behaviour preservation of a biological regulatory network when embedded into a larger network, Fundam. Inform. 91 (2009) 463–485. doi:10.3233/FI-2009-0052.

[26] M. Mabrouki, M. Aiguier, J.-P. Comet, P. Gall, A. Richard, Embedding of biological regulatory networks and property preservation, Mathematics in Computer Science 5 (2011) 263–288. doi:10.1007/s11786-011-0092-3.

[27] F. Delaplace, H. Klaudel, T. Melliti, S. Sené, Analysis of modular organisation of interaction networks based on asymptotic dynamics, in: D. Gilbert, M. Heiner (Eds.), Computational Methods in Systems Biology, LNCS 7605, Springer, 2012, pp. 148–165.

[28] K. Perrot, P. Perrotin, S. Sené, A framework for (de)composing with Boolean automata networks, in: J. Durand-Lose, S. Verlan (Eds.), MCU 2018, LNCS 10881, Springer, 2018, pp. 121–136.

[29] R. Rudell, A. Sangiovanni-Vincentelli, Multiple-valued minimization for pla optimization, IEEE Transactions on Computer-Aided Design CAD-6 (5) (1987) 727–750.